PROJECT ON MACHINE LEARNING

BUSINESS REPORT

Sankesh A Nagrare
Sankeshnagrare92@gmail.com
PGP – Data Science & Business Analytics

Case Study 1 – Linear Regression

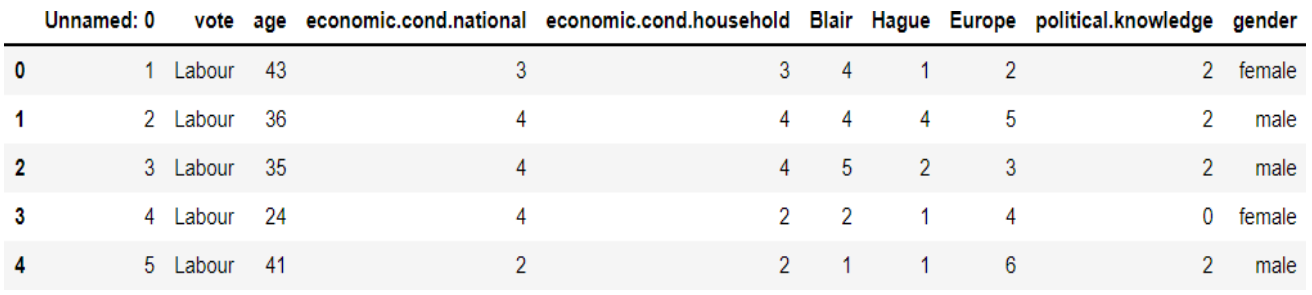
Overview:

You are hired by one of the leading news channels CNBE who wants to analyze recent elections. This survey was conducted on 1525 voters with 9 variables. You have to build a model, to predict which party a voter will vote for on the basis of the given information, to create an exit poll that will help in predicting overall win and seats covered by a particular party.

Summary:

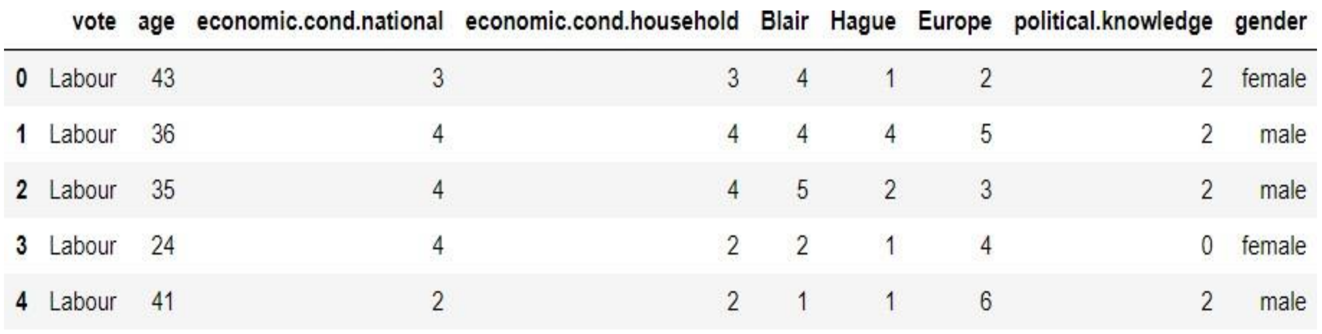
This business report provides detailed explanation on the approach to each problem definition, solution to those the problems provide some key insights/recommendations to the business.

Q1.1) Read the dataset. Do the descriptive statistics and do the null value condition check. Write an inference on it.



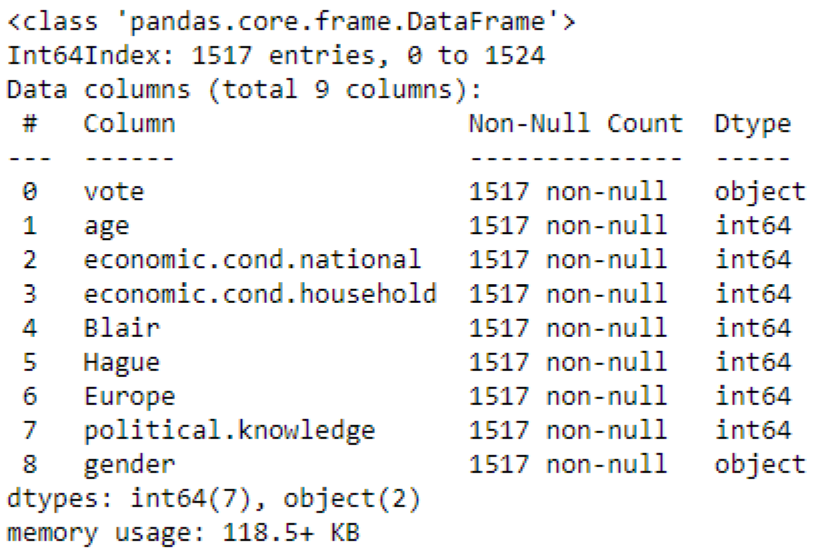
	Unnamed: 0	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
0	1	Labour	43	3	3	4	1	2	2	female
1	2	Labour	36	4	4	4	4	5	2	male
2	3	Labour	35	4	4	5	2	3	2	male
3	4	Labour	24	4	2	2	1	4	0	female
4	5	Labour	41	2	2	1	1	6	2	male

Original Sample of the dataset



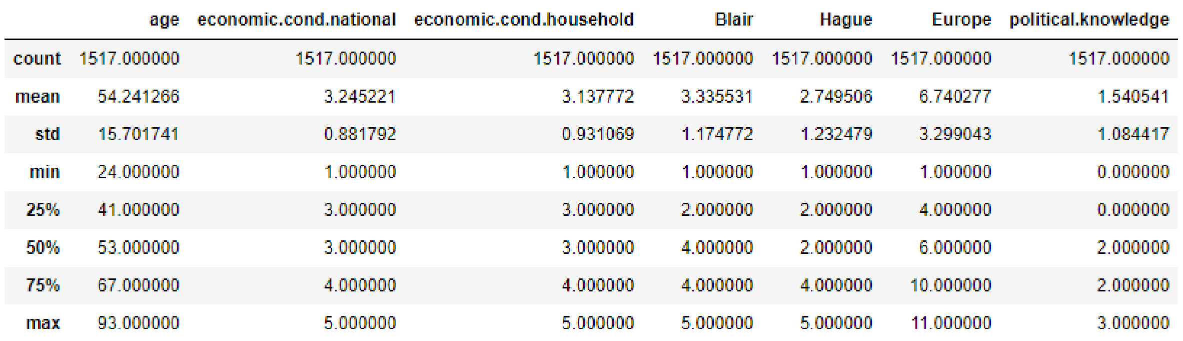
	vote	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender
0	Labour	43	3	3	4	1	2	2	female
1	Labour	36	4	4	4	4	5	2	male
2	Labour	35	4	4	5	2	3	2	male
3	Labour	24	4	2	2	1	4	0	female
4	Labour	41	2	2	1	1	6	2	male

Sample of the dataset after removal of irrelevant columns



```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1517 entries, 0 to 1524
Data columns (total 9 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   vote                     1517 non-null   object
 1   age                      1517 non-null   int64 
 2   economic.cond.national   1517 non-null   int64 
 3   economic.cond.household  1517 non-null   int64 
 4   Blair                    1517 non-null   int64 
 5   Hague                    1517 non-null   int64 
 6   Europe                   1517 non-null   int64 
 7   political.knowledge      1517 non-null   int64 
 8   gender                   1517 non-null   object
dtypes: int64(7), object(2)
memory usage: 118.5+ KB
```

Basic Information of the dataset



	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge
count	1517.000000	1517.000000	1517.000000	1517.000000	1517.000000	1517.000000	1517.000000
mean	54.241266	3.245221	3.137772	3.335531	2.749506	6.740277	1.540541
std	15.701741	0.881792	0.931069	1.174772	1.232479	3.299043	1.084417
min	24.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000
25%	41.000000	3.000000	3.000000	2.000000	2.000000	4.000000	0.000000
50%	53.000000	3.000000	3.000000	4.000000	2.000000	6.000000	2.000000
75%	67.000000	4.000000	4.000000	4.000000	4.000000	10.000000	2.000000
max	93.000000	5.000000	5.000000	5.000000	5.000000	11.000000	3.000000

Table 4: Summary of the dataset

Check for Duplicates

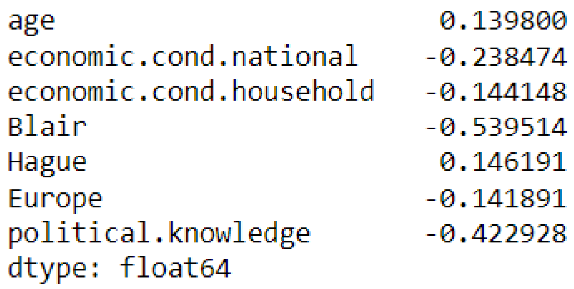
Below is the output of python code -

```
Number of duplicate rows = 8
```

Below is the number of rows and columns before and after removal of duplicate rows

```
Before (1525, 9)
After (1517, 9)
```

Skewness Check of variables



```
age                         0.139800
economic.cond.national     -0.238474
economic.cond.household    -0.144148
Blair                      -0.539514
Hague                       0.146191
Europe                     -0.141891
political.knowledge        -0.422928
dtype: float64
```

Variable skewness

Value Counts for all variables

```
AGE :  70
93     1
91     1
90     1
92     2
87     3
      ..
54    37
47    38
35    38
49    39
37    42
Name: age, Length: 70, dtype: int64
```

```
VOTE :  2
Conservative     460
Labour          1057
Name: vote, dtype: int64
```

```
ECONOMIC.COND.NATIONAL :  5
1     37
5     82
2    256
4    538
3    604
Name: economic.cond.national, dtype: int64
```

```
GENDER :  2
male      709
female    808
Name: gender, dtype: int64
```

```
ECONOMIC.COND.HOUSEHOLD :  5
1     65
5     92
2    280
4    435
3    645
Name: economic.cond.household, dtype: int64
```

```
EUROPE :  11
2      77
7      86
10    101
1     109
8     111
9     111
5     123
4     126
3     128
6     207
11    338
Name: Europe, dtype: int64
```

```
BLAIR :  5
3      1
1     97
5    152
2    434
4    833
Name: Blair, dtype: int64
```

```
POLITICAL.KNOWLEDGE :  4
1     38
3    249
0    454
2    776
Name: political.knowledge, dtype: int64
```

```
HAGUE :  5
3     37
5     73
1    233
4    557
2    617
Name: Hague, dtype: int64
```

Observations -

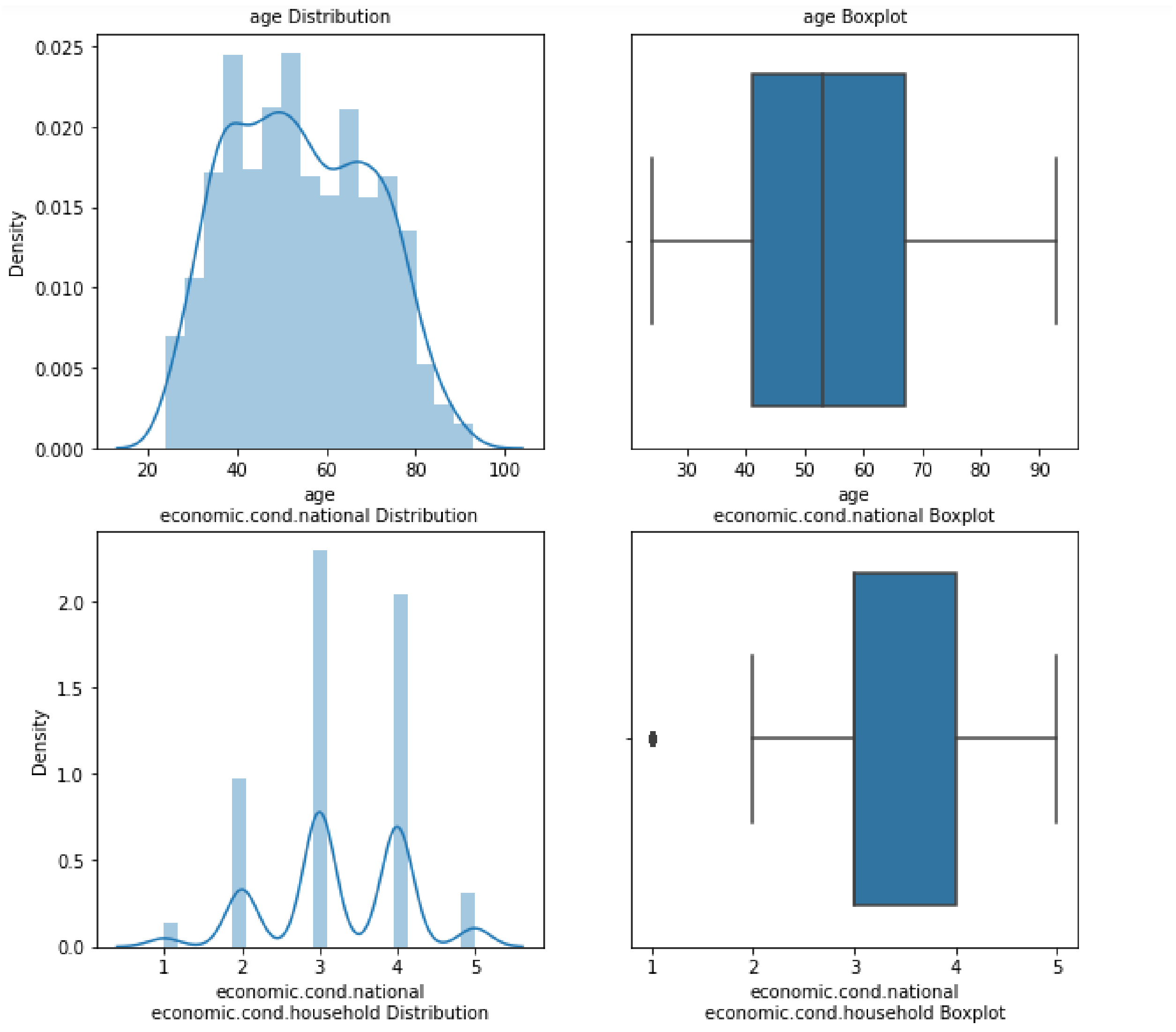
- The dataset contains 1525 rows and 9 columns before removal of duplicate rows.
- The variables/columns 'vote' & 'gender' both are of object datatypes, whereas all the other variables are of integer datatype.
- The dataset does not contain any null values & dirty or false values in the dataset.
- The dataset contains 8 duplicate rows. We can remove them as they are very less compared to the entire dataset and they don't add any value to the study.
- The dataset now contains 1517 rows and 9 columns after removal of duplicate rows.
- The variables 'vote' & 'gender' both contains 2 unique values each.
- The mean age of all the voters is 54, the 25th & 75h percentile of voter's age is 41 & 67 respectively. The minimum and maximum age of a voter is 24 & 93 respectively.
- Since, the skewness value of variable 'Blair' is between 0.5 and +1, it shows moderately skewed distribution.
- Since, the skewness value of all the remaining variables is between -0.5 & +0.5, they all show approximately symmetric distribution.
- From the data dictionary and the dataset we come to know that except for variable 'age' all the other variables are of categorical nature.

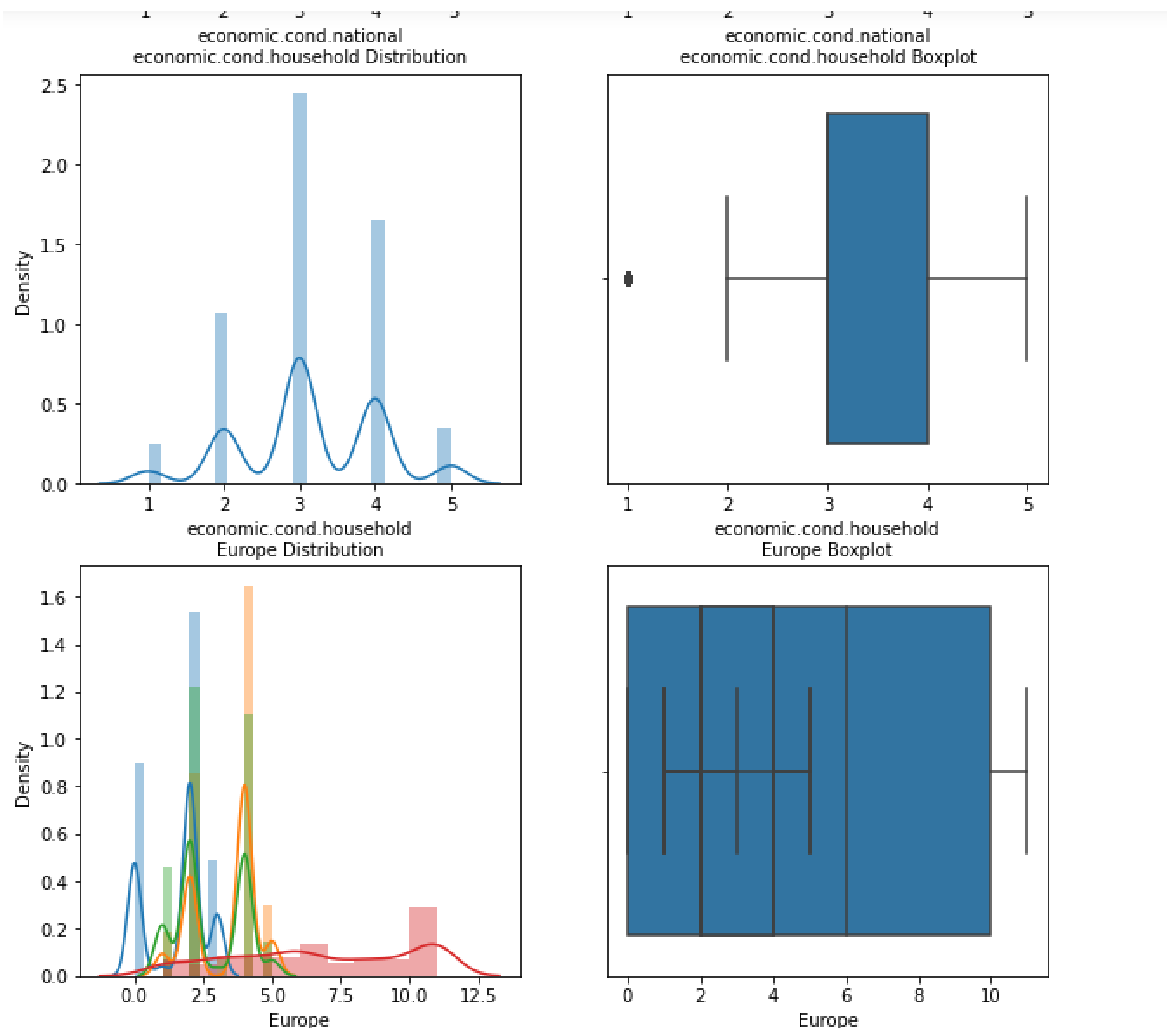
Q1.2) Perform Univariate and Bivariate Analysis. Do exploratory data analysis. Check for Outliers.

Univariate Analysis

1) Outlier Identification

2) Distribution check





Boxplot for Outlier Identification And Distribution check

From the figure above, we can say that,

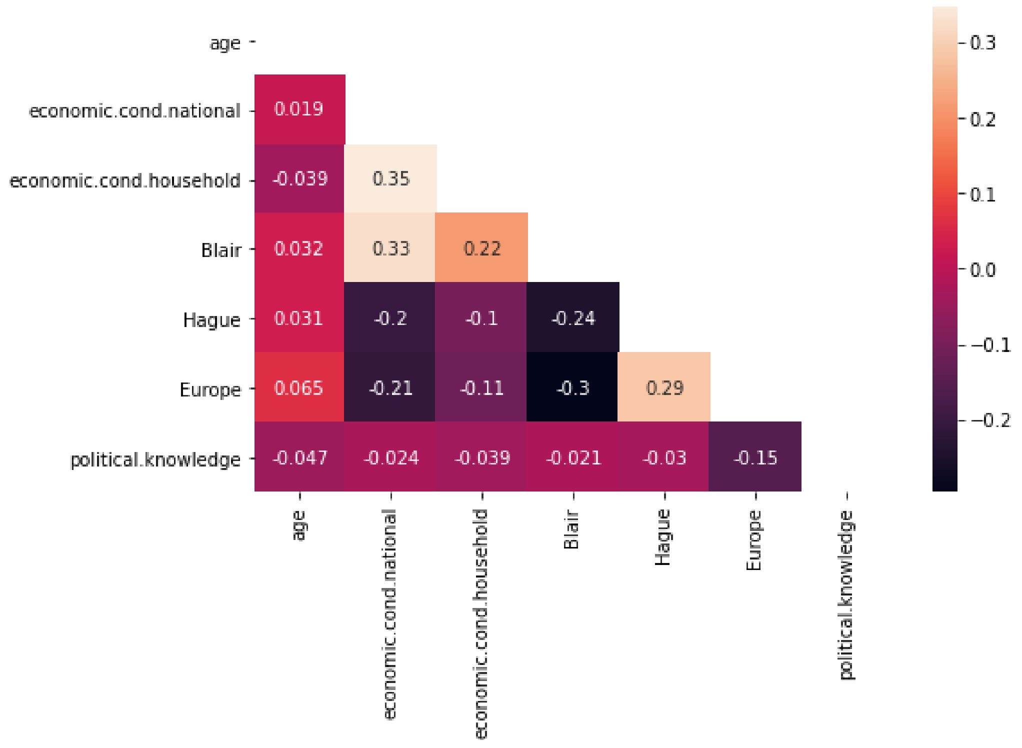
- The dataset does contain very few outliers.
- Treating outliers sometimes results in the models having better performance but the models lose out on generalization. Hence, we won't be treating them in order to not lose out on generalization.

From the figure above, we can say that,

- The distribution plot summarizes the results of the skew values in a visual form, which makes it easier for us to understand.
- The variable 'Blair' shows moderately skewed distribution.
- The remaining variables show approximately symmetric distribution.

Multivariate Analysis

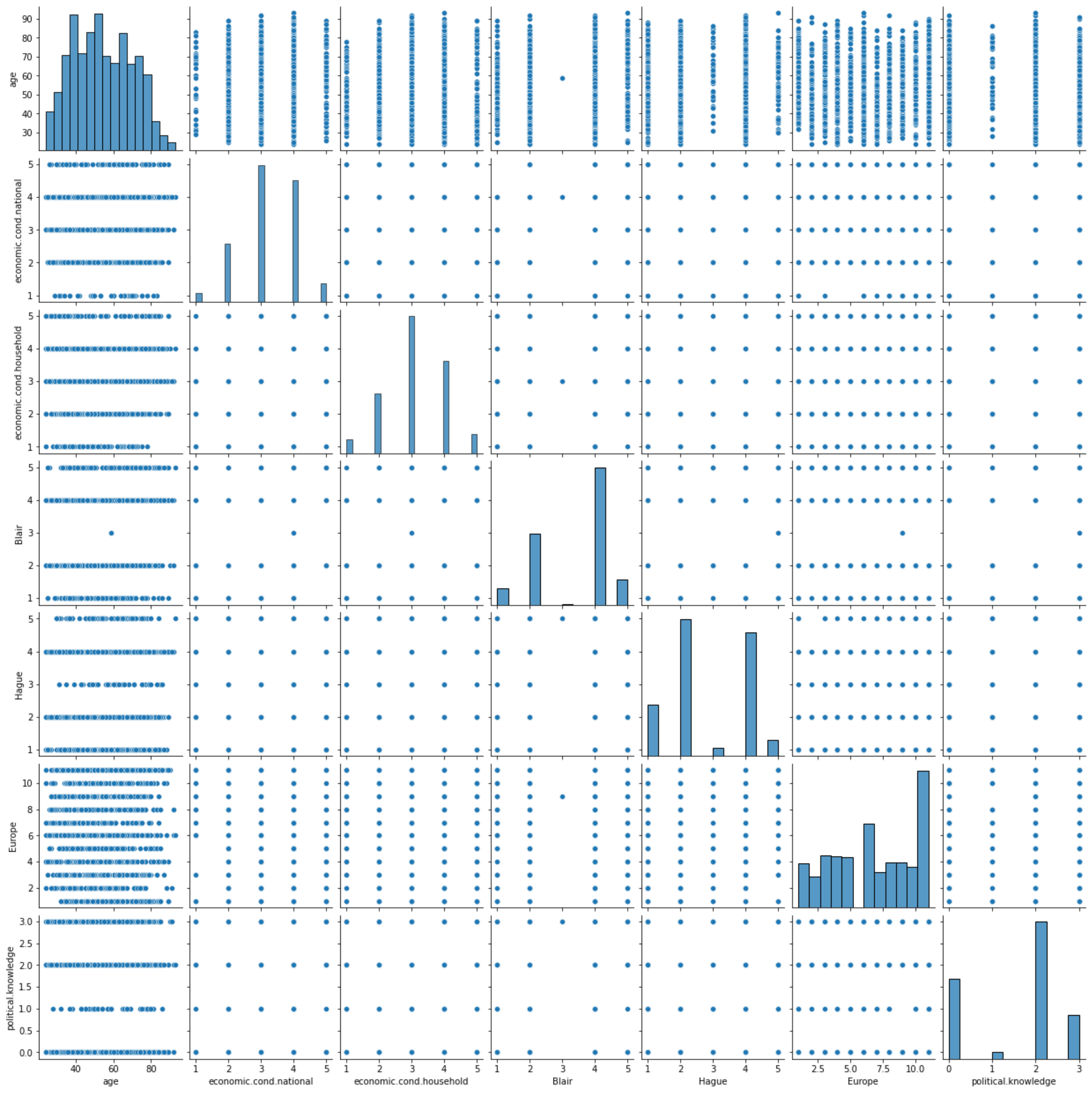
1) Heatmap to study correlation between variables



Correlation Matrix

From the figure above, we can say that,

- There is no strong correlation among any variable combinations.
- All of them show very weak or no correlation at all.

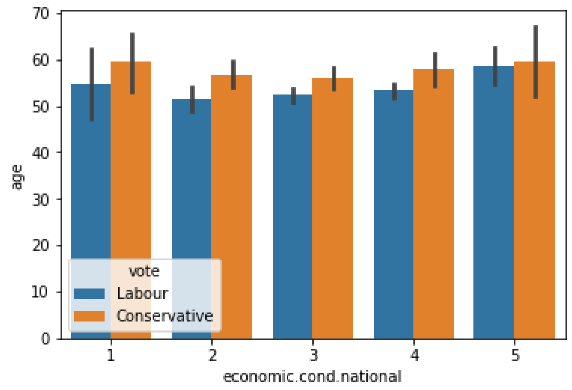


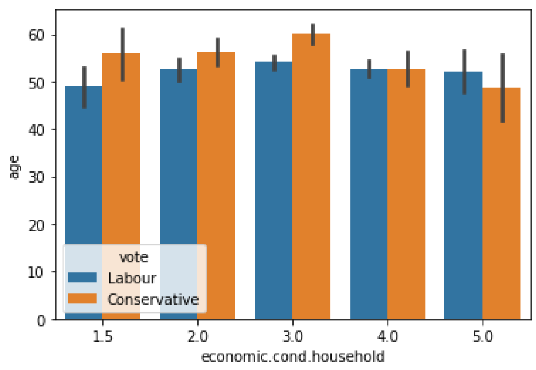
Pair plot for all variable combinations

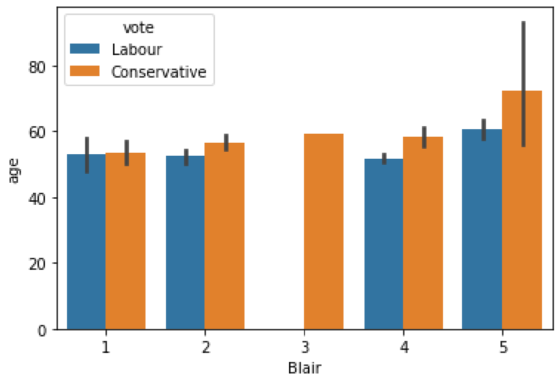
From the figure above, we can say that,

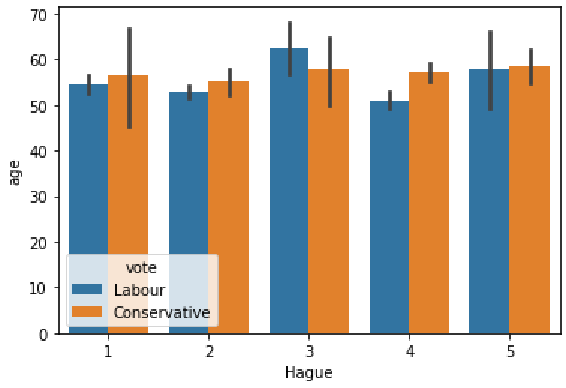
- Pairplot visualises the correlation heatmap in a x vs y graph, which also shows weak or no corrleation among the variables.

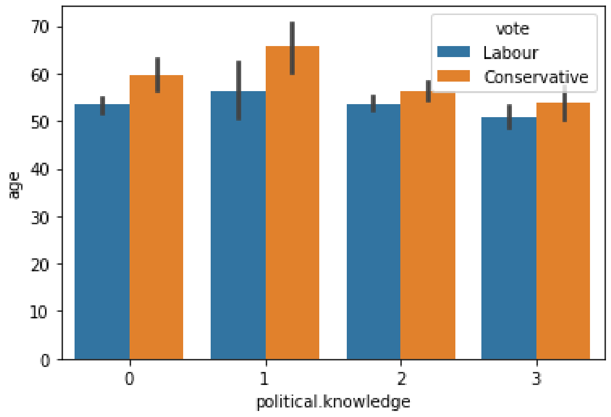
Understanding Target variable 'price' with different Categorical Variables

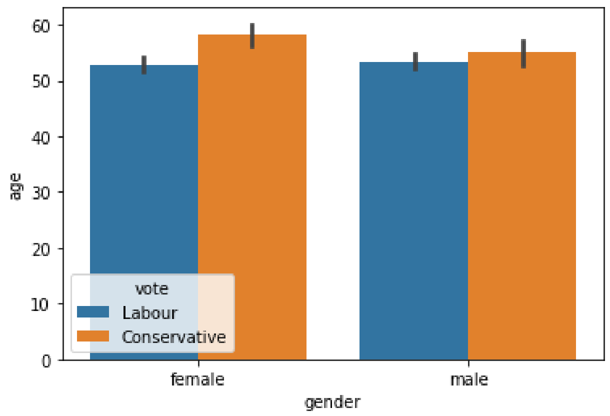










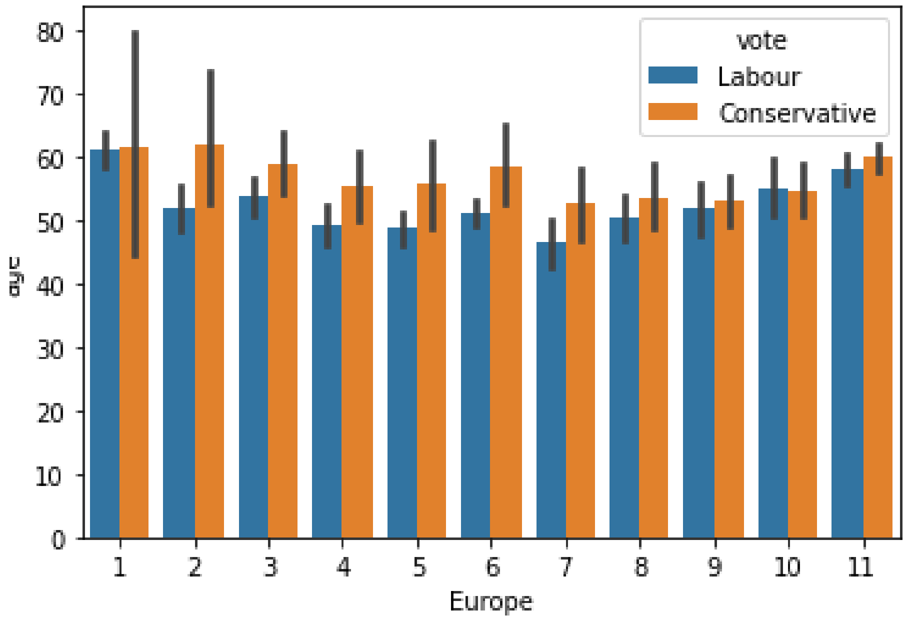


All the numerical variables except for age have definite unique values, hence, they can be treated as categorical or object datatype to understand the data in much better fashion.

Understanding Target variable 'vote' with different Categorical Variables

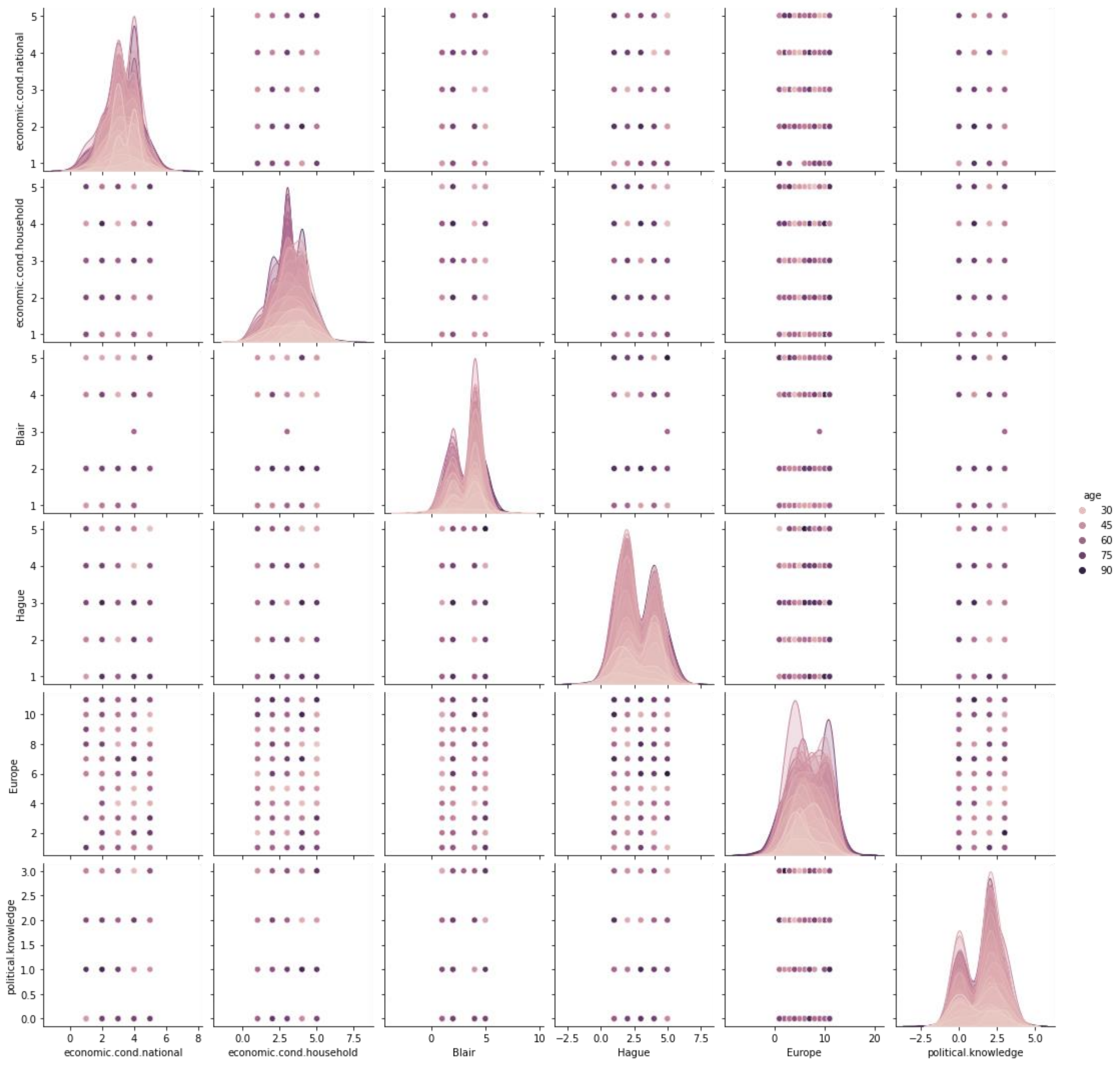
From the graph above, we can say that,

- The average age of voters who vote for
- Conservative party is higher compared to Labour party for both males & females



.

- The average age of voters who vote for
  - Conservative party is higher compared to Labour party for all the assessment levels of current national economic conditions.
- The average age of voters who vote for
  - Conservative party is higher compared to Labour party for the assessment levels '1', '2' & '3' of current household economic conditions.
  - Approximately same for assessment level '4' of current household economic conditions.
  - Labour Party is higher compared to Conservative party for assessment levels '5' of current household economic conditions.
- The average age of voters who vote for
  - Conservative party is higher compared to Labour party for all the assessment levels of assessment of Labour leader.
- The average age of voters who vote for
  - Conservative party is higher compared to Labour party for the assessment levels '1', '2', '4' & '5' of assessment of Conservative leader.
  - Labour party is higher compared to Conservative party for assessment level '3' of assessment of conservative leader.
- The average age of voters who vote for
  - Conservative party is higher compared to Labour party for all the assessment levels of knowledge of parties' positions on European integration.
- The average age of voters who vote for
  - Conservative party is higher compared to Labour party for the assessment levels '2' to '9' & '11' of assessment of Eurosceptic sentiment.
  - Approximately same for assessment level '1' of Euroscpetic sentiment.
  - Labour party is higher compared to Conservative party for assessment level '10' of Eurosceptic sentiment.

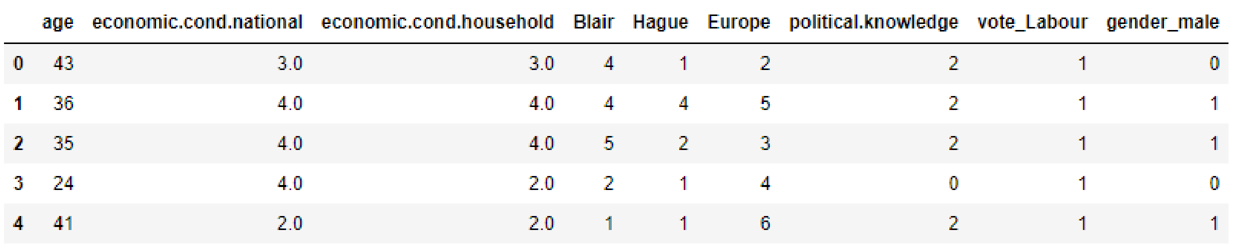


Pairplot to understand target variable much better with respect to other variables

Q1.3) Encode the data (having string values) for Modelling. Is Scaling necessary here or not? Data Split: Split the data into train and test (70:30)

Encoding Object datatypes

- We encode variables having string values such as 'vote' & 'gender' for modelling.
- We do Label encoding for our target variable 'vote' & One hot encoding for remaining variables.
- We don't do label encoding for remaining variables as they might lead to a problem. Since, there are different numbers in the same column, the model might misunderstand the data to be in some kind of order, 0 < 1.



	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	vote_Labour	gender_male
0	43	3.0	3.0	4	1	2	2	1	0
1	36	4.0	4.0	4	4	5	2	1	1
2	35	4.0	4.0	5	2	3	2	1	1
3	24	4.0	2.0	2	1	4	0	1	0
4	41	2.0	2.0	1	1	6	2	1	1

Dataset after encoding of categorical variables

Is scaling necessary or not?

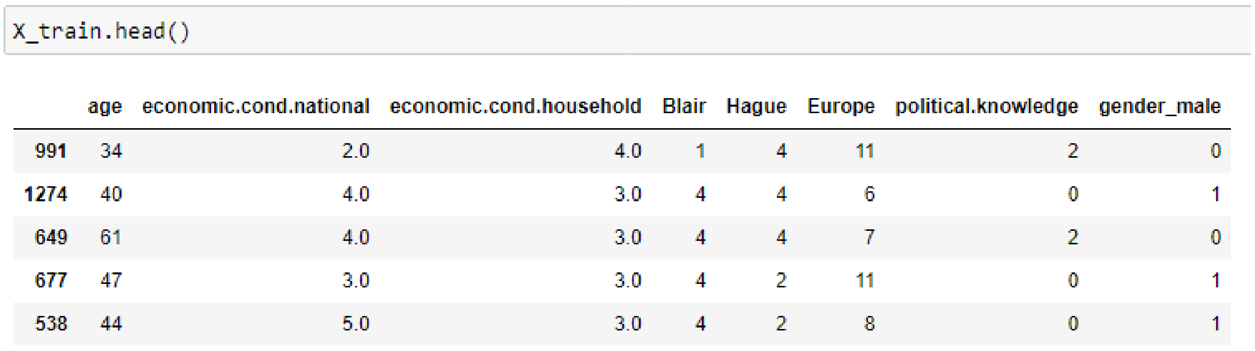
- Scaling, in this particular case won't be making much of a difference because 'age' is the only numerical variable.
- All the other remaining variables represent categories for assessment levels for different parameters.

Therefore, Scaling here is not necessary or rather not required.

There is no issue of class imbalance here as we have reasonable proportions in both the classes.

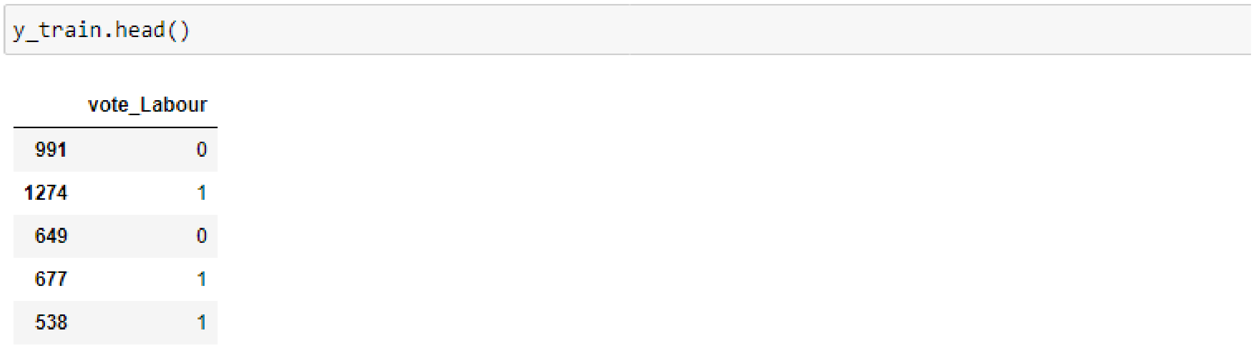
Data Split: Splitting data into training and testing data

Extracting the target column into separate vectors for training set and test set.



```
X_train.head()
```

	age	economic.cond.national	economic.cond.household	Blair	Hague	Europe	political.knowledge	gender_male
991	34	2.0	4.0	1	4	11	2	0
1274	40	4.0	3.0	4	4	6	0	1
649	61	4.0	3.0	4	4	7	2	0
677	47	3.0	3.0	4	2	11	0	1
538	44	5.0	3.0	4	2	8	0	1

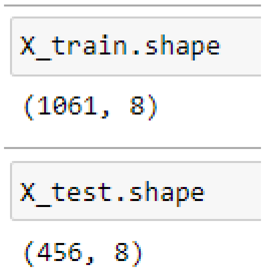


```
y_train.head()
```

	vote_Labour
991	0
1274	1
649	0
677	1
538	1

Once, we extract the variables into separate vectors, we split the dataset into 70:30 ratio for training and test respectively

Dimensions of the dataset after splitting which is 70:30 (training:test)



```
X_train.shape
```

```
(1061, 8)
```

```
X_test.shape
```

```
(456, 8)
```

Q1.4) Apply Logistic Regression and LDA (linear discriminant analysis).

Logistic Regression Model

1) Initially we use the default parameters to see how the model performs.
Default parameters:

- Penalty = 12
- Solver = Ibfgs
- Max_iter = 10000

```
LogisticRegression(max_iter=10000, n_jobs=2, penalty='none', solver='newton-cg',
                   verbose=True)
```

-

2) Then we fit the model into the data using the above parameters.
3) Once done, we then predict for the training and testing dataset (will be used further for model accuracy, classification report & confusion matrix).
4) And then, we predict the probability for the training and testing dataset (will be used further for AUC & ROC score).
5) Afterwards, we find accuracy for Logistic regression model

```
Model Accuracy for training dataset is : 0.8341187558906692
Model Accuracy for  Testing dataset is : 0.8289473684210527
```

Linear Discriminant Analysis Model

1) Initially we use the default parameters to see how the model performs.
Default parameters:
Solver = svd
Shrinkage = none

```
GridSearchCV(cv=3, estimator=LogisticRegression(max_iter=1000
0, n_jobs=2),
             n_jobs=-1,
             param_grid={'penalty': ['l2', 'none'], 'solver':
['sag', 'lbfgs'],
                         'tol': [0.0001, 1e-05]},
             scoring='f1')

{'penalty': 'l2', 'solver': 'lbfgs', 'tol': 0.0001}

LogisticRegression(max_iter=10000, n_jobs=2)
```

2) Then we fit the model into the data using the above parameters.
3) Once done, we then predict for the training and testing dataset (will be used further for model accuracy, classification report & confusion matrix).
4) And then, we predict the probability for the training and testing dataset (will be used further

for AUC & ROC score).
5) Afterwards, we find accuracy for Linear Discriminant Analysis model

```
Model Accuracy for training dataset is : 0.8341187558906692

Model Accuracy for  Testing dataset is : 0.831140350877193
```

Observations –

- Both the models (Logistic regression & LDA) have good accuracies for both training and test dataset.
- As the difference in accuracies of training and test dataset is very small, the models are good performing and valid models.

Q1.5) Apply KNN Model and Naïve Bayes Model. Interpret the results.

K-nearest neighbors model

1) Initially we use the default parameters to see how the model performs.
Default parameters:

- n_neighbors = 5
- weights = uniform
- algorithm = auto

2) Then we fit the model into the data using the above parameters.
3) Once done, we then predict for the training and testing dataset (will be used further for model accuracy, classification report & confusion matrix).
4) And then, we predict the probability for the training and testing dataset (will be used further for AUC & ROC score).
5) Afterwards, we find accuracy for K-nearest neighbors model

```
Model Accuracy for training dataset is : 0.8539114043355325
Model Accuracy for  Testing dataset is : 0.8157894736842105
```

Naive Bayes model

1) Initially we use the default parameters to see how the model performs.
Default parameters:

```
GaussianNB()
```

2) Then we fit the model into the data using the above parameters.
3) Once done, we then predict for the training and testing dataset (will be used further for model accuracy, classification report & confusion matrix).
4) And then, we predict the probability for the training and testing dataset (will be used further for AUC & ROC score).
5) Afterwards, we find accuracy for Naive Bayes model

```
Model Accuracy for training dataset is:0.8341187558906692
Model Accuracy for  Testing dataset is:0.8223684210526315
```

Observations -

- Both the models (K-Nearest Neighbours & Naive Bayes) have good accuracies for both training and test dataset.
- As the difference in accuracies of training and test dataset is very small, the models are good

performing and valid models.

Q1.6) Model Tuning, Bagging (Random Forest should be applied for Bagging), and Boosting.

Model Tuning
In order to do so, we use GridSeachCV to find best possible parameter combination for each model.

For Logistic Regression model
The parameters used to find the optimum ones are
penalty – none
solver – newton-cg,
max_iter - 10000
The value of used random state = 1.
Best Parameters:

```
LogisticRegression(max_iter=10000, n_jobs=2, penalty='none', solver='newton-cg',
                   verbose=True)
```

For Linear Discriminant Analysis model
The parameters used to find the optimum ones are
penalty –none, auto
solver – lsqr,
The value of used random state = 1.
Best Parameters:

```
{'shrinkage': 'auto', 'solver': 'lsqr'}
```

```
LinearDiscriminantAnalysis(shrinkage='auto', solver='lsqr')
```

For K-nearest neighbors model
The parameters used to find the optimum ones are
n_neighbors – 5,6,7,8,
n_neighbors -9,10,11,12,13
weights – uniform
algorithm - bruteBest Parameters:

```
{'algorithm': 'brute', 'n_neighbors': 9, 'weights': 'uniform'}
```

```
KNeighborsClassifier(algorithm='brute', n_neighbors=9)
```

Observations -
Except for Logistic Regression model, all the other models have slightly different parameters as best parameters compared to the default ones.

Model Accuracies for all models with best parameters into consideration-

1) Logistic Regression model

```
 Model Accuracy for training dataset is : 0.8341187558906692
  Model Accuracy for  Testing dataset is : 0.8289473684210527
```

2) Linear Discriminant Analysis model

```
Model Accuracy for training dataset is : 0.8341187558906692
Model Accuracy for  Testing dataset is : 0.8289473684210527
```

3) K-nearest neighbors model

```
Model Accuracy for training dataset is : 0.8539114043355325
```
Model Accuracy for testing dataset is : 0.8157894736842105

4) Naive Bayes model

```
 Model Accuracy for training dataset is:0.8341187558906692
Model Accuracy for testing dataset is: 0.8223684210526315
```

Observations –
Accuracies for the models have very slightly changed by performing model tuning. All the models are good for prediction as they have higher accuracies.

Bagging model

1) We create a bagging model using random forest as its estimator and random state =1.

```
BaggingClassifier(base_estimator=DecisionTreeClassifier(), n_estimators=100,
                  random_state=1)
```

2) Then we fit the model into the data using the above parameters.

3) Once done, we then predict for the training and testing dataset (will be used further for model accuracy, classification report & confusion matrix).

4) And then, we predict the probability for the training and testing dataset (will be used further for AUC & ROC score).

5) Afterwards, we find accuracy for bagging model.

```
Model accuracy for training dataset is: 0.9670122525918945
Model accuracy for test dataset is: 0.8289473684210527
```

Boosting model (Ada Boost)

1) We create a boosting model using random state =1.

2) Then we fit the model into the data.

```
AdaBoostClassifier(n_estimators=100, random_state=1)
```

3) Once done, we then predict for the training and testing dataset (will be used further for model accuracy, classification report & confusion matrix).

4) And then, we predict the probability for the training and testing dataset (will be used further for AUC & ROC score).

5) Afterwards, we find accuracy for boosting model.

```
Model Accuracy for training dataset is: 1.0
```

Model Accuracy for training dataset is: 0.8201754385964912

Observations -

- Bagging model has a high difference in accuracy for training and test data, meaning it is poor performing.
- Ada Boost model has good accuracy for both training and test data. It is a good model with high accuracy.

Q1.7) Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model. Final Model: Compare the models and write inference which model is best/optimized.

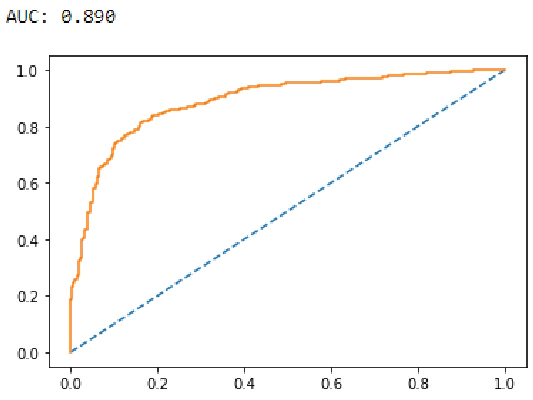
Performance metrics of Logistic Regression model

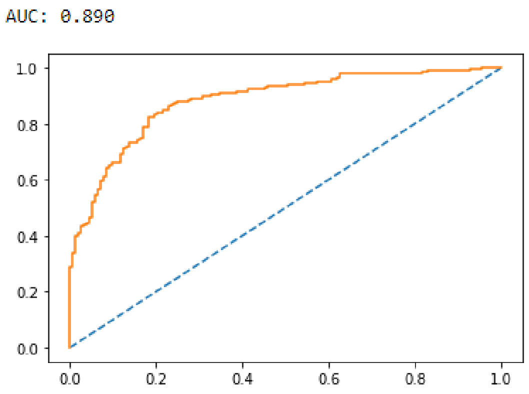
Since, the model parameters for Logistic Regression model turned out to be same before and after tuning. Hence, we could go ahead with either for comparing it with other models.

1) Model Accuracy

```
Model Accuracy for training dataset is : 0.8341187558906692
Model Accuracy for  Testing dataset is : 0.8289473684210527
```

2) AUC & ROC Curve for training and testing data





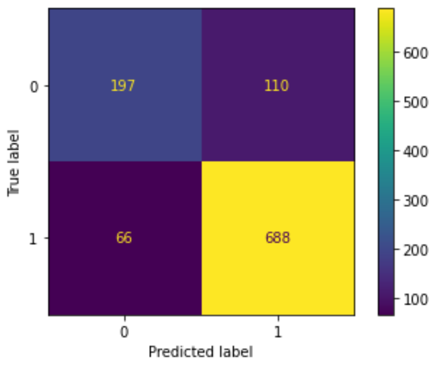
3)

AUC & ROC curve for training and test data for LR model

4) Confusion matrix for training and test data

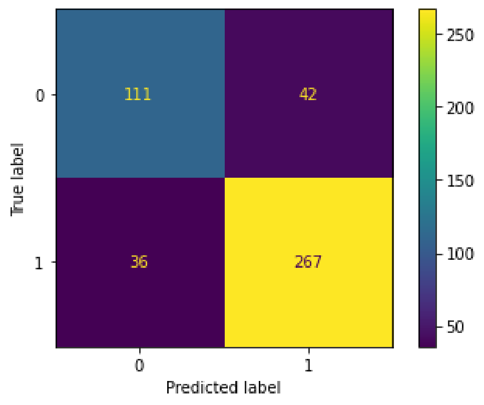
1)Confusion Matrix For training data set

```
array([[197, 110],
       [ 66, 688]], dtype=int64)
```



2)Confusion Matrix For testing data set

```
array([[111,  42],
       [ 36, 267]], dtype=int64)
```



5) Classification Report

```
Classification Report of the training data:

              precision    recall  f1-score   support

           0       0.74      0.65      0.69       307
           1       0.86      0.91      0.89       754

    accuracy                           0.83      1061
   macro avg       0.80      0.78      0.79      1061
weighted avg       0.83      0.83      0.83      1061


Classification Report of the test data:

              precision    recall  f1-score   support

           0       0.76      0.73      0.74       153
           1       0.86      0.88      0.87       303

    accuracy                           0.83       456
   macro avg       0.81      0.80      0.81       456
weighted avg       0.83      0.83      0.83       456
```

Logistic Regration Conclusion

1)Training data set

```
AUC for the Training Data: 0.890
AUC for the Test Data: 0.888
```

Accuracy - Training Data : `0.8341187558906692`

Accuracy - Testing Data: `0.831140350877193`

Performance metrics of Linear Discriminant Analysis model

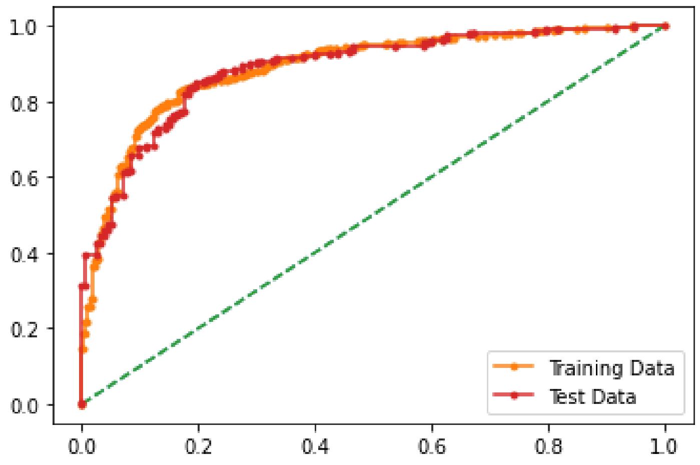
Since, the model parameters for Linear Discriminant Analysis model turned out to be better for default parameters compared to the tuning ones. Hence, we could go ahead with the default one for comparing it with other models.

1) Model Accuracy

2) Accuracy - Training Data : `0.8341187558906692`

3) Accuracy - Testing Data: `0.831140350877193`

4) AUC & ROC curve



AUC & ROC curve for training and test data for LDA model

5) Confusion matrix for training and test data

Confusion matrix of the training data :

```
[[307   0]
 [  0 754]]
```

Confusion matrix of the testing data :

```
[[108  45]
 [ 37 266]]
```

6) Classification report for training and test data

```
Classification Report of the training data:

              precision    recall  f1-score   support

           0       0.74      0.65      0.69       307
           1       0.86      0.91      0.89       754

    accuracy                           0.83      1061
   macro avg       0.80      0.78      0.79      1061
weighted avg       0.83      0.83      0.83      1061


Classification Report of the test data:

              precision    recall  f1-score   support

           0       0.76      0.73      0.74       153
           1       0.86      0.88      0.87       303

    accuracy                           0.83       456
   macro avg       0.81      0.80      0.81       456
weighted avg       0.83      0.83      0.83       456
```

Linear Discriminant Analysis Conclusion

Train Data:

AUC:89%

Accuracy :83%
Precision :86%
F1-Score : 89%

Test Data:

AUC:88%

Accuracy :83%
Precision :86%
F1-Score : 87%

Performance metrics of K-nearest neighbors model

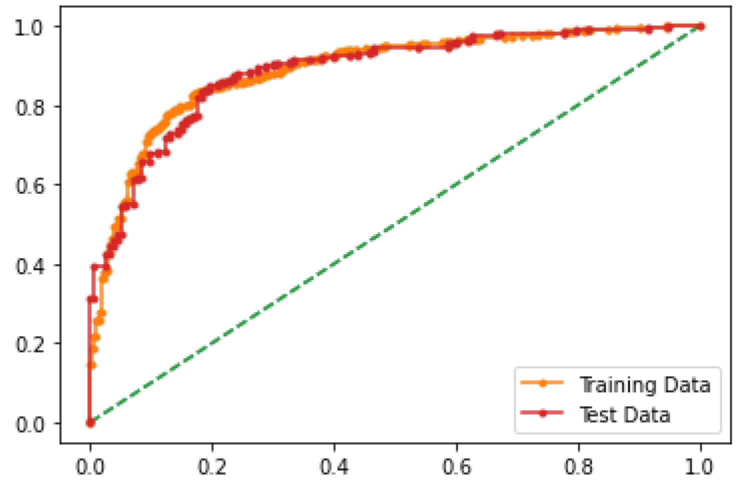
Since, the model parameters for K-nearest neighbors model turned out to be better for default parameters compared to the tuning ones. Hence, we could go ahead with the default one for comparing it with other models.

1) Model accuracy

```
Model accuracy for training data set :  0.8341187558906692
Model accuracy for testing data set : 0.8157894736842105
```

2) AUC & ROC curve for training and test data

```
AUC for the Training Data: 0.890
AUC for the Test Data: 0.888
```



3) Confusion Matrix for training and test data

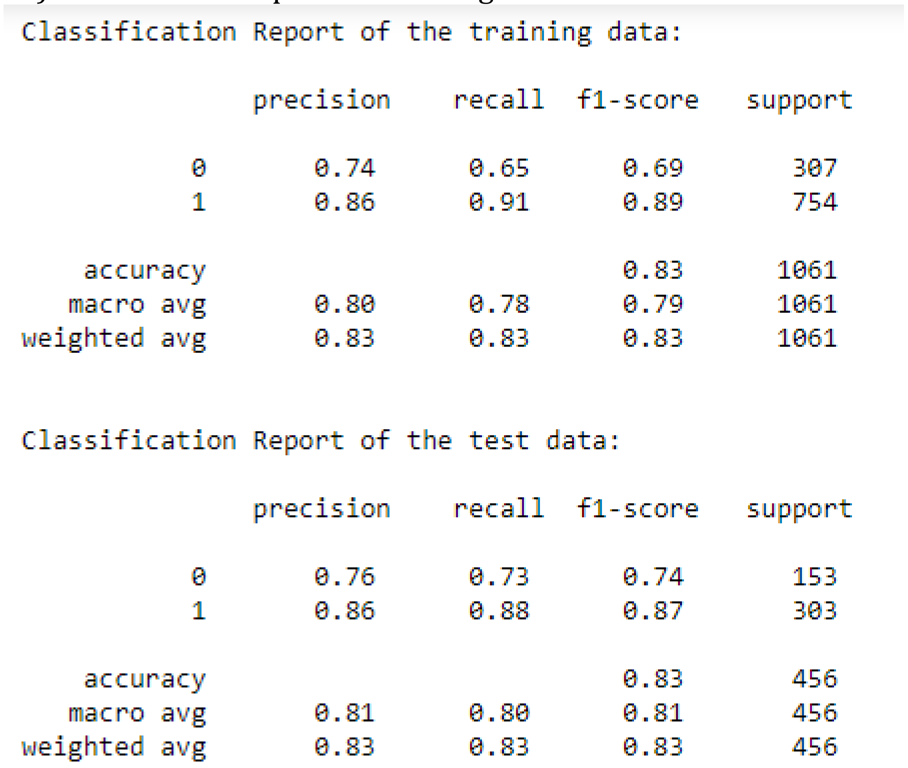
Training Data

```
0.8341187558906692
[[212  95]
 [ 81 673]]
```

Testing data

```
0.8157894736842105
[[ 99  54]
 [ 30 273]]
```

4) Classification report for training and test data



```
Classification Report of the training data:

              precision    recall  f1-score   support

           0       0.74      0.65      0.69       307
           1       0.86      0.91      0.89       754

    accuracy                           0.83      1061
   macro avg       0.80      0.78      0.79      1061
weighted avg       0.83      0.83      0.83      1061


Classification Report of the test data:

              precision    recall  f1-score   support

           0       0.76      0.73      0.74       153
           1       0.86      0.88      0.87       303

    accuracy                           0.83       456
   macro avg       0.81      0.80      0.81       456
weighted avg       0.83      0.83      0.83       456
```

Performance metrics of Naive Bayes model

Since, the model parameters for Naive Bayes model turned out to be better for best parameters compared to the default ones. Hence, we could go ahead with the tuned model for comparing it with other models.

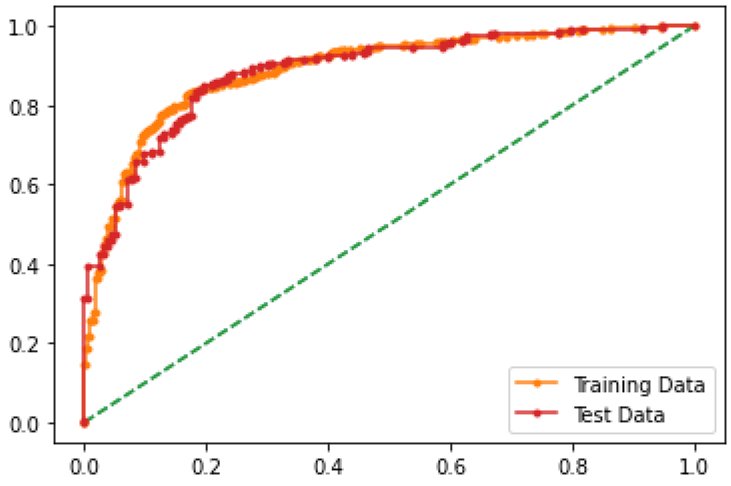
1) Model accuracy

```
Model accuracy for training data set :  0.8341187558906692

Model accuracy for testing data set : 0.8223684210526315
```

2) AUC & ROC curve for training and test data

```
AUC for the Training Data: 0.890
AUC for the Test Data: 0.888
```



AUC & ROC curve for training and test data for NB model

4) Confusion matrix for training and test data

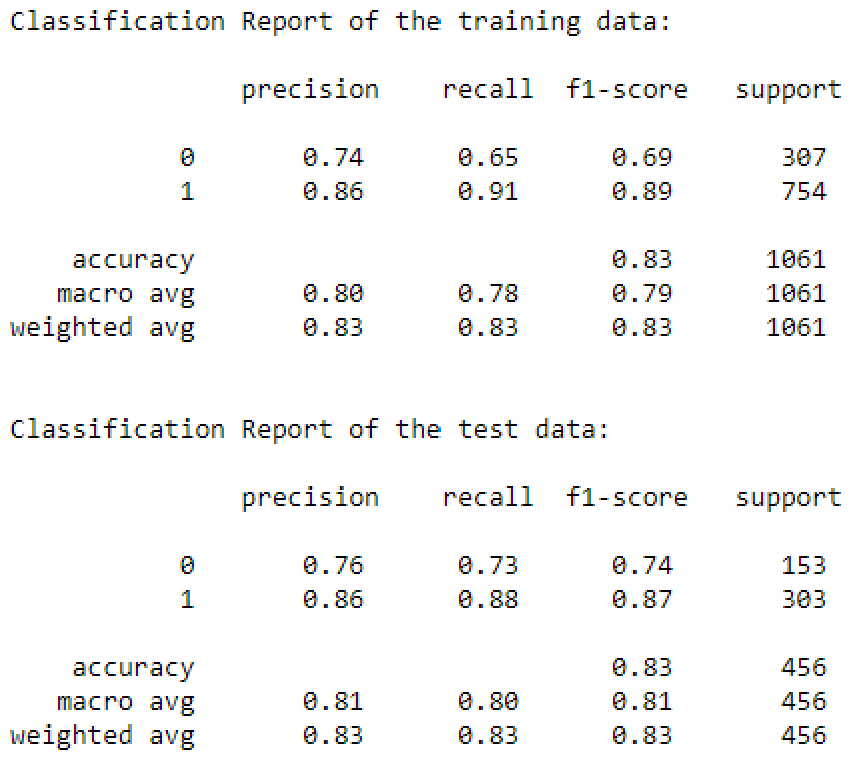
Confusion matrix Train data :

```
[[212  95]
 [ 81 673]]
```

Confusion matrix Test data :

```
[[112  41]
 [ 40 263]]
```

5) Classification report for training and test data



```
Classification Report of the training data:

              precision    recall  f1-score   support

           0       0.74      0.65      0.69       307
           1       0.86      0.91      0.89       754

    accuracy                           0.83      1061
   macro avg       0.80      0.78      0.79      1061
weighted avg       0.83      0.83      0.83      1061


Classification Report of the test data:

              precision    recall  f1-score   support

           0       0.76      0.73      0.74       153
           1       0.86      0.88      0.87       303

    accuracy                           0.83       456
   macro avg       0.81      0.80      0.81       456
weighted avg       0.83      0.83      0.83       456
```

Performance metrics of Bagging model

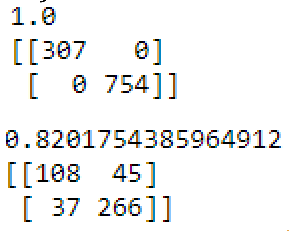
1) Model accuracy

Model accuracy for training data set :1.0

```
Model accuracy for testing  data set : 0.8201754385964912
```

2) AUC & ROC curve for training and test data

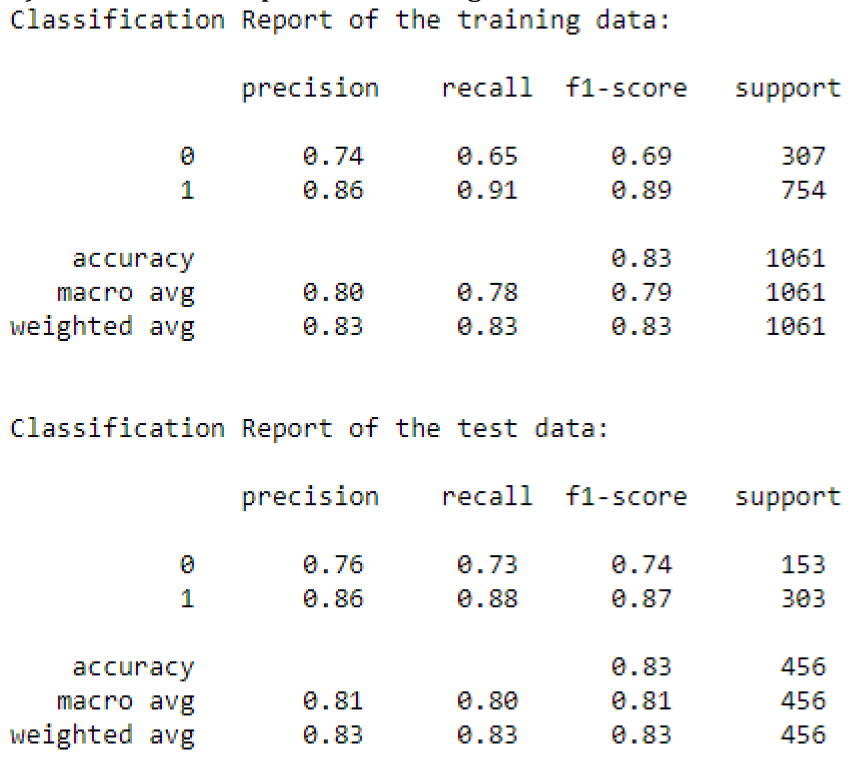
3) Confusion matrix for training and test data



```
1.0
[[307   0]
 [  0 754]]

0.8201754385964912
[[108  45]
 [ 37 266]]
```

4) Classification report for training and test data



```
Classification Report of the training data:

              precision    recall  f1-score   support

           0       0.74      0.65      0.69       307
           1       0.86      0.91      0.89       754

    accuracy                           0.83      1061
   macro avg       0.80      0.78      0.79      1061
weighted avg       0.83      0.83      0.83      1061


Classification Report of the test data:

              precision    recall  f1-score   support

           0       0.76      0.73      0.74       153
           1       0.86      0.88      0.87       303

    accuracy                           0.83       456
   macro avg       0.81      0.80      0.81       456
weighted avg       0.83      0.83      0.83       456
```

Performance metrics of Boosting model (Ada Boost)

1) Model accuracy

Train data Accuracy :0.8501413

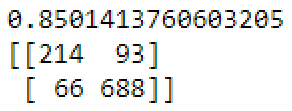
Test data accuracy :0.81359612

2) AUC & ROC Curve for training and test data

UC & ROC curve for training and test data for Boosting model

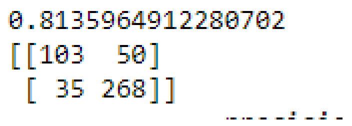
3) Confusion matrix for training and test data

Train data



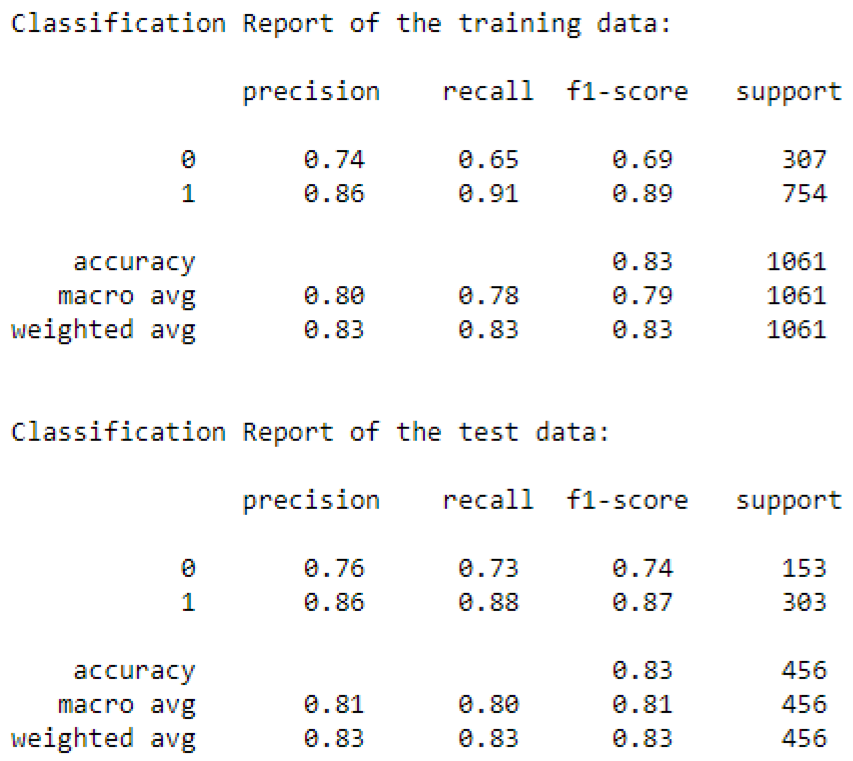
```
0.8501413760603205
[[214  93]
 [ 66 688]]
```

Test data



```
0.8135964912280702
[[103  50]
 [ 35 268]]
```

4) Classification report for training and test data



```
Classification Report of the training data:

              precision    recall  f1-score   support

           0       0.74      0.65      0.69       307
           1       0.86      0.91      0.89       754

    accuracy                           0.83      1061
   macro avg       0.80      0.78      0.79      1061
weighted avg       0.83      0.83      0.83      1061


Classification Report of the test data:

              precision    recall  f1-score   support

           0       0.76      0.73      0.74       153
           1       0.86      0.88      0.87       303

    accuracy                           0.83       456
   macro avg       0.81      0.80      0.81       456
weighted avg       0.83      0.83      0.83       456
```

Feature Importance
Conclusion -

- Considering the comparison table of performance metrics for all the models we can say that,
  - All the models are good performing models except for Bagging, as bagging seems to be an over-fitted model.
  - All the other models can be used for prediction purposes, as they all have higher accuracies, AUC & ROC score, f1-score, precision and recall.
- Considering the comparison graph of ROC score for training and test data for all the models, we can say that, bagging is the best model for prediction as it has all the higher values of accuracy AUC & ROC score, f1-score, precision and recall.
- Since, other models don't have feature importance feature, from LR & LDA model we can say that,
  - The most important variable is Blair (Assessment of Labour party leader).
  - Then comes the variable age (how old the voter is).
  - Then comes economic.conditions.household (assessment of current household economic conditions).

Q1.8) Based on these predictions, what are the insights?

Insights -

From the analysis above, we can say that

- All the models except for bagging model are good performing models and any of them can be used for the predictions of exit poll for the next elections, as bagging model seems to look like an over-fitted model.
- In general, older people seem more likely to vote for conservative party, whereas younger people seem more likely to vote for labour party.
- Current household conditions play a vital role in determining which party a voter will vote for.
- Also, leaders of labour party are more likely to drive voting towards a conclusive decision.
- Gender does not play much of a role in voter's voting decision.
- People who do not support European integration tend to vote for Labour party and people who do support European integration tend to vote for conservative party.
- Given, what we know, Labour party is more likely to win the elections as per predictions by the model.

Case Study 2- Text Mining

Overview:

In this particular project, we are going to work on the inaugural corpora from the nltk in Python. We will be looking at the following speeches of the Presidents of the United States of America:

President Franklin D. Roosevelt in 1941
President John F. Kennedy in 1961
President Richard Nixon in 1973

Summary:

This business report provides detailed explanation on the approach to each problem definition, solution to those the problems provide some key insights/recommendations to the business.

Q2.1) Find the number of characters, words, and sentences for the mentioned documents.